

Object Containers in EViews

B. Essama-Nssah
Poverty Reduction Group (PRMP)
The World Bank
Module 1

Outline

- Introduction
- Workfile
- Database

Introduction

- EViews stands for Econometric Views, a Windows application designed by Quantitative Micro Software (QMS) for general econometric analysis and model simulation.
- The language is structured around the notion of object, a collection of related information and operations.
- EViews offers both an interactive interface and a batch mode of command execution.

Introduction

- The current version provides a wide range of practical tools for statistical analysis, estimation (e.g. OLS, 2SLS, 3SLS, GMM, MLE, FMLE, discrete and limited dependent variable models, panel estimation).
- The available simulation tools can also be used to build and solve general equilibrium models (CGEs).
- The same tools make it possible to build and run macro-micro simulation models for the study of the poverty and distributional impact of macroeconomic shocks and policies.

Introduction

- All objects are held in workfiles or in databases (focus of this presentation).
- First step in any EViews session involves the creation of a new workfile or loading in the memory an existing one.

Workfile

- The purpose of a workfile is to hold objects during processing.
- Its structure is therefore determined by the nature of such objects.
- Objects designed to hold the contents of datasets (hence the type of data to analyze) ultimately determine the structure of a workfile or workfile page.

Workfile

- A dataset is a collection of observations on one or more variables.
- An observation is a measurement of some manifestation of a phenomenon.
- Observations are made over time or across space, therefore there is a need to uniquely identify each observation in the dataset.

Workfile

- Identifiers provide information about the observed phenomenon, the date or place of observation, or the individual entity to which the observation pertains.
- Variable name identifies the observed phenomenon.
- For annual data about a process, one can use year identifiers such as "2001", "2002" etc...

Workfile

- For cross-sectional data on household, for instance, a code for household ID uniquely identifies a record in a dataset.
- For longitudinal or panel data, one uses both a cross-section ID and a date ID to identify each observation.

Workfile

- Observation identifiers are used to describe the structure of a workfile.
- Patterns for dated regular frequency data:
 - Annual
 - Semi-annual
 - Quarterly
 - Monthly
 - Weekly
 - Daily (5 day week)
 - Daily (7 day week)

Workfile

- Undated data
 - Considered as unstructured.
 - Use default integer identifiers (1, 2, 3, . . . , n) that enumerate observations in the dataset.
- A balanced panel represents a regular frequency panel data structure.
 - [Panel data implies each observation has both a group or cross-section (e.g. country) and a cell or within group (e.g. year) identifiers.]

Workfile

- Examples of workfile creation:
 - `wfcreate(wf=kenya, page=polak_model) a 1968 1977`
 - `wfcreate(wf=semiannual, page=data) s 05:1 06:2`
 - `wfcreate(wf=semiannual, page=data) s 2005s1 2006s2`
 - `Wfcreate(wf=macrofile,page=qdata) q 1995q1 2005q4`
 - `wfcreate(wf=macrofile, page=qdata) q 95:1 05:4`
 - `wfcreate(wf=monthfile, page=monthdata) m 49:02
05:02`
 - `wfcreate(wf=monthfile, page=monthdata) m 1949m2
2005m2`
 - `wfcreate(wf=india, page=rural) u 13`

(s1 stands for first semester, s2 for second)

Workfile

- Creating a workfile
 - The command **WFCREATE** creates a workfile by describing its structure.
 - Syntax for regular dated data:
 - **WFCREATE(OPTIONS) FREQUENCY
START_DATE END_DATE**
 - Options provide names for workfile and workfile page:
WFCREATE(WF=WF_NAME, PAGE=PAGE_NAME)

Workfile

- A panel data set has a balanced structure when each group (or cross-sectional unit) has the same regular frequency containing the same date observations.
- A file to contain such data can be created in two steps.

Workfile

- Create a standard workfile that could hold time series for each cross-sectional unit. The range of this file should include the earliest and latest dates pertaining to any cross-section.
- Provide information on the number of cross-sections
- `WFCREATE(WF=DATAFILE, PAGE=BPANEL) Q
1970Q1 2020Q4 200`

Workfile

- To create a new page:

- `PAGECREATE(PAGE=PAGE_NAME)
FREQUENCY START_DATE END_DATE
[NUM_CROSS_SECTIONS]`

- `PAGECREATE(PAGE=PAGE_NAME) U
NUM_OBS (Number of observations)`

- To open workfile:

- `WFOPEN[PATH\]WORKFILE_NAME`

Workfile

- To save:
 - `WFSAVE(OPTIONS)`
`[PATH\]WORKFILE_NAME`
 - `OPTIONS:`
 - `1` Single precision
 - `2` Double precision
 - `C` Compressed
- To make a workfile active:
 - `WFSELECT`
`WORKFILE_NAME[\PAGE_NAME]`

Workfile

- Workfile Window
 - Provides access to:
 - All available data via a directory for the objects in a given workfile page.
 - Tools for working with workfiles and their pages.
 - Various elements:
 - Title bar shows workfile designation "**WORKFILE**" followed by the name of the workfile.
 - The full disk path if file has been saved on disk.

Workfile

- Workfile Window (continued)
 - Various elements:
 - Button bar for ease of access to useful workfile operations.
 - Two lines of status information below the tool bar where EViews displays the range of the workfile, the current sample, and the display filter i.e. a rule used to select a subset of objects to display in the workfile window.
 - The main portion of the window shows the workfile directory for the contents of the workfile.

Workfile

- Workfile Window (continued)
 - Various elements:
 - View/Details +/- button can be used to toggle between the standard workfile display format and a display of additional information about the date the objects were created or updated, and the label information.
 - The Workfile Summary View provides a description of the current workfile structure, along with the list of the types and numbers in each of the workfile page.

Database

- Like a workfile, a database is an object container.
- Key differences with the workfile:
 - No need to load up the database into memory in order to access content.
 - Objects can be stored to or retrieved directly from the database on disk.
 - Series in the database need not have the same frequency or range.

Database

- Structure
 - Capacity limit: 1 million objects per database.
 - A set of files on disk consisting of a main file with extension .EDB and a series of index files with following extensions .E0, .E1A, .E1B.

Database

- To create a database:
 - `DBCREATE DB_NAME`
 - `DB DB_NAME`
 - Example: `DB SHOCKS`
- To open existing database
 - `DB DB_NAME` or `DBOPEN DB_NAME`

Database

- To put objects in database use STORE command:
 - STORE OBJ1 OBJ2 OBJ3 OBJ4 ...
- To retrieve objects, use FETCH
 - FETCH OBJ1 OBJ2 OBJ3 OBJ4 ...
- It is possible to make direct use of series contained in the database without first retrieving them
 - Syntax: DB_NAME::OBJECT_NAME



The End.